# Deep Generative Modelling

## Project Presentation for CS772A

Archit Sharma (14129)
Abhinav Agrawal (14011)
Anubhav Shrivastava (14114)

# Table of contents

# Introduction

# Introduction

- Classic task of estimating the density function.
- Objectives in context of Deep Learning context slightly relaxed: Generate realistic samples and provide likelihood measurements.
- Advances in Deep Learning have provided a real jump in our capacity to model and generate from multi-modal high-dimensional distributions, particularly those associated with natural images.

# Motivation

- Represents our ability to manipulate high dimensional spaces, and extract meaningful representations
- Extremely important in context of Semi-supervised Learning (in general when training data available is low) or when we have missing data
- Naturally handles Multi-modal outputs

# Some Recent Approaches

Two game changing works:

- **Variational Autoencoders**: Explicit density measurement with approximate posterior maximization.
- **Generative Adversarial Networks**: Implicit Density Maximization.

Some other frameworks based on Maximum Likelihood Estimation: Real NVP, PixelRNN.
We looked at many frameworks: Generative Latent Optimization (GLO), Bayesian GANs, Normalizing and Inverse Autoregressive Flows.

# Background

# Normalizing Flows

Traditionally, variational inference employs simple families of posterior approximations to allow efficient inference. With the help of normalizing flows, a simple initial density is transformed into a density of desired complexity by applying a sequence of transformations.

- Suppose $z$ has a distribution $q(z)$ and $z' = f(z)$, then distribution of z' is given by:

$$q(z') = q(z)\left|det\left(\frac{\partial f^{-1}}{\partial z'}\right)\right| = q(z)\left|det\left(\frac{\partial f}{\partial z}\right)\right|^{-1}$$

- Above mentioned simple maps can be combined several times to construct complex densities:

$$z_K = f_K \bigcirc \ldots \bigcirc f_2 \bigcirc f_1(z_0)$$

$$\ln q_K(z_K) = \ln q_0(z_K) - \Sigma_{k=1}^{K} \ln \left|det\frac{\partial f_k}{\partial z_{k-1}}\right|$$

# Normalizing Flows

- Normalizing flows proposes to use the following transformation:

$$f(z) = z + \mathbf{u}h(w^T x + b)$$

- The determinant of the Jacobian:

$$\psi(z) = h'(w^T z + b)\mathbf{w}$$

$$\left| det \frac{\partial f}{\partial z} \right| = |det(\mathbf{I} + \mathbf{u}^T \psi(z)^T)| = |1 + \mathbf{u}^T \psi(z)|$$

- We can apply a sequence of above transformations to get $q_K$:

$$\ln q_K(z_K) = \ln q_0(z_K) - \Sigma_{k=1}^{K} \ln |1 + \mathbf{u}_k^T \psi_k(z_{k-1})|$$

# Real NVP

Real NVP transformations is a framework for doing invertible and efficiently learnable transformations, leading to an unsupervised learning algorithm with exact log-likelihoods, efficient sampling and inference of latent variables.

- Change of variable formula:

$$p_X(x) = p_Z(f(x))\left|det\left(\frac{\partial f(x)}{\partial x}\right)\right|$$

- Coupling layers:

$$y_{1:d} = x_{1:d}$$

$$y_{d+1:D} = x_{d+1:D} \odot exp\big(s(x_{1:d})\big) + t(x_{1:d})$$

- Jacobian of the above transformation is a lower triangular matrix which reduces the computation cost for its calculation.

# Real NVP

- The above transformation is invertible:

$$x_{1:d} = y_{1:d}$$

$$x_{d+1:D} = \Big(y_{d+1:D} - t(y_{1:d})\Big) \odot exp\Big(-s(y_{1:d})\Big)$$

- The above mentioned transformations leaves some of the components unchanged. The **coupling layers** can be composed in alternate fasion to solve this issue.

# Approach

# Approach

Normalizing Flows provides a framework, incorporated within VAEs, where more complex posteriors can be obtained by using invertible transformations. The constraint on the transformation: *Determinant of Jacobian matrix should be efficiently computable.* We propose to

use Real NVP transformations. These transformations are much more powerful than those proposed in Normalizing flows, but at the same time have efficient Jacobian computation as well.

# Framework Details

We look to model the Binarized MNIST. The model structure is similar to those in VAEs and Normalizing Flows.

- *Encoder*: Passes images through a set of convolutional and pooling layers. Then uses a few fully connected layes to convert each image into a fixed size embedding.
- *Transformations*: In lines with Normalizing Flows, the embedding from the encoder is passed through a sequence of real NVP transformations
- *Decoder*: The transformed embedding is converted into an image by passing through a sequence of transposed convolutional layers.

Each transformation is a a set of ”coupling layers”, such that no dimension of the embedding is untransformed

# Optimization Objective

$$
\begin{aligned}
F(x) &= \mathbb{E}_{q_\phi(z|x)}[\log q_\phi(z|x) - \log p(x,z)] \\
&= \mathbb{E}_{q_0(z_0)}[\log q_K(z_K) - \log p(x,z_K)] \\
F(x)_{NF} &= \mathbb{E}_{q_0(z_0)}[\log q_0(z_0)] - \mathbb{E}_{q_0(z_0)}[\log p(x,z_K)] \\
&- \mathbb{E}_{q_0(z_0)}\left[\Sigma_{k=1}^{K} \ln |1 + \mathbf{u}_k^T \psi_k(z_{k-1})|\right] \\
F(x)_{rNVP} &= \mathbb{E}_{q_0(z_0)}[\log q_0(z_0)] - \mathbb{E}_{q_0(z_0)}[\log p(x,z_K)] \\
&- \mathbb{E}_{q_0(z_0)}\left[\Sigma_{k=1}^{K} (s_{1,k}(b \odot z_{k-1}))\right]
\end{aligned}
$$

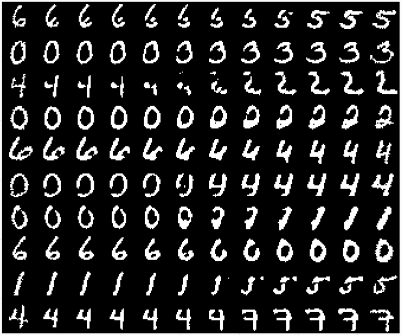
# Results

# Results

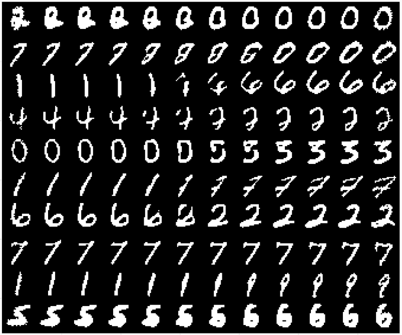
Table 1: **NF**: Normalizing Flows, **rNVP**: Real NVP. k denotes the number of transformations

| Models | log p(x\|z) |
|---|---|
| rNVP (k = 2) | 60.57 |
| rNVP (k = 5) | 75.56 |
| rNVP (k = 10) | 75.01 |
| rNVP (k = 20) | 81.37 |
| NF (k = 4) | 65.5 |
| NF (k =10 ) | 68.9 |
| NF (k = 20) | 75.4 |
| NF (k = 40) | 83.4 |

# Latent Space Interpolation



(a) Latent Space Interpolations for a simple VAE



(b) Latent Space interpolations with rNVP transformations

Figure 1

# Ongoing Work

# Ongoing Work

- Implement the above algorithms to larger datasets (SVHN, CIFAR10, CelebA)
- Include better stabilization techniques such as weight normalization and batch normalization in realNVP transformations to train deeper networks
- Experiment with Convolutional Neural Networks for transformations ($s$ and $t$ can both be any arbitrary transformation)

Questions?